INTERNATIONAL STANDARD

ISO/IEC 14496-3

Second edition
2001-12-15

Information technology — Coding of audio-visual objects —

Part 3: **Audio**

Technologies de l'information — Codage des objets audiovisuels —

Partie 3: Codage audio

Reference number
ISO/IEC 14496-3:2001(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

ISO/IEC 14496-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This second edition cancels and replaces the first edition (ISO/IEC 14496-3:1999), which has been technically revised. It incorporates Amd.1:2000 and Cor.1:2001.

ISO/IEC 14496 consists of the following parts, under the general title *Information technology — Coding of audio-visual objects*:

— *Part 1: Systems*

— *Part 2: Visual*

— *Part 3: Audio*

— *Part 4: Conformance testing*

— *Part 5: Reference software*

— *Part 6: Delivery Multimedia Integration Framework (DMIF)*

— *Part 7: Optimized software for MPEG-4 visual tools*

— *Part 8: Carriage of MPEG-4 contents over IP networks*

Annexes 2.E, 3.C, 4.A and 5.A form a normative part of this part of ISO/IEC 14496. Annexes 1.A to 1.C, 2.A to 2.D, 3.A, 3.B, 3.D to 3.F, 4.B, 5.B to 5.F, 6.A and 7.A are for information only.

Due to its technical nature, this part of ISO/IEC 14496 requires a special format as several standalone electronic files and, consequently, does not conform to some of the requirements of the ISO/IEC Directives, Part 2.

Introduction

Overview

ISO/IEC 14496-3 (MPEG-4 Audio) is a new kind of audio standard that integrates many different types of audio coding: natural sound with synthetic sound, low bitrate delivery with high-quality delivery, speech with music, complex soundtracks with simple ones, and traditional content with interactive and virtual-reality content. By standardizing individually sophisticated coding tools as well as a novel, flexible framework for audio synchronization, mixing, and downloaded post-production, the developers of the MPEG-4 Audio standard have created new technology for a new, interactive world of digital audio.

MPEG-4, unlike previous audio standards created by ISO/IEC and other groups, does not target a single application such as real-time telephony or high-quality audio compression. Rather, MPEG-4 Audio is a standard that applies to *every* application requiring the use of advanced sound compression, synthesis, manipulation, or playback. The subparts that follow specify the state-of-the-art coding tools in several domains; however, MPEG-4 Audio is more than just the sum of its parts. As the tools described here are integrated with the rest of the MPEG-4 standard, exciting new possibilities for object-based audio coding, interactive presentation, dynamic soundtracks, and other sorts of new media, are enabled.

Since a single set of tools is used to cover the needs of a broad range of applications, *interoperability* is a natural feature of systems that depend on the MPEG-4 Audio standard. A system that uses a particular coder—for example a real-time voice communication system making use of the MPEG-4 speech coding toolset—can easily share data and development tools with other systems, even in different domains, that use the same tool—for example a voicemail indexing and retrieval system making use of MPEG-4 speech coding.

The remainder of this clause gives a more detailed overview of the capabilities and functioning of MPEG-4 Audio. First a discussion of concepts, that have changed since the MPEG-2 audio standards, is presented. Then the MPEG-4 Audio toolset is outlined.

New concepts in MPEG-4 Audio

Many concepts in MPEG-4 Audio are different than those in previous MPEG Audio standards. For the benefit of readers who are familiar with MPEG-1 and MPEG-2 we provide a brief overview here.

- **MPEG-4 has no standard for transport.** In all of the MPEG-4 tools for audio and visual coding, the coding standard ends at the point of constructing a sequence of access units that contain the compressed data. The MPEG-4 Systems (ISO/IEC 14496-1:2001) specification describes how to convert the individually coded objects into a bitstream that contains a number of multiplexed sub-streams.

There is no standard mechanism for transport of this stream over a channel; this is because the broad range of applications that can make use of MPEG-4 technology have delivery requirements that are too wide to easily characterize with a single solution. Rather, what is standardized is an interface (the Delivery Multimedia Interface Format, or DMIF, specified in ISO/IEC 14496-6:1999) that describes the capabilities of a transport layer and the communication between transport, multiplex, and demultiplex functions in encoders and decoders. The use of DMIF and the MPEG-4 Systems bitstream specification allows transmission functions that are much more sophisticated than are possible with previous MPEG standards.

However, LATM and LOAS were defined to provide a low overhead audio multiplex and transport mechanism for natural audio applications, which do not require sophisticated object-based coding or other functions provided by MPEG-4 Systems.

The following table gives an overview about the multiplex, storage and transmission formats for MPEG-4 Audio currently available within the MPEG-4 framework:

	Format	Functionality defined in:	Functionality redefined in:	Description
Multiplex	FlexMux	ISO/IEC 14496-1:2001 (MPEG-4 Systems) (Normative)	-	Flexible multiplex scheme
	LATM	ISO/IEC 14496-3:2001 (MPEG-4 Audio) (Normative)	-	Low Overhead Audio Transport Multiplex
Storage	ADIF	ISO/IEC 13818-7:1997 (MPEG-2 Audio) (Normative)	ISO/IEC 14496-3:2001 (MPEG-4 Audio) (Informative)	(MPEG-2 AAC) Audio Data Interchange Format, AAC only
	MP4FF	ISO/IEC 14496-1:2001 (MPEG-4 Systems) (Normative)	-	MPEG-4 File format
Transmission	ADTS	ISO/IEC 13818-7:1997 (MPEG-2 Audio) (Normative, Exemplarily)	ISO/IEC 14496-3:2001 (MPEG-4 Audio) (Informative)	Audio Data Transport Stream, AAC only
	LOAS	ISO/IEC 14496-3:2001 (MPEG-4 Audio) (Normative, Exemplarily)	-	Low Overhead Audio Stream, based on LATM, three versions are available: AudioSyncStream() EPAudioSyncStream() AudioPointerStream()

To allow for a user on the remote side of a channel to dynamically control a server streaming MPEG-4 content, MPEG-4 defines backchannel streams that can carry user interaction information.

- **MPEG-4 Audio supports low-bitrate coding.** Previous MPEG Audio standards have focused primarily on transparent (undetectable) or nearly transparent coding of high-quality audio at whatever bitrate was required to provide it. MPEG-4 provides new and improved tools for this purpose, but also standardizes (and has tested) tools that can be used for transmitting audio at the low bitrates suitable for Internet, digital radio, or other bandwidth-limited delivery. The new tools specified in MPEG-4 are the state-of-the-art tools that support low-bitrate coding of speech and other audio.

- **MPEG-4 is an object-based coding standard with multiple tools.** Previous MPEG Audio standards provided a single toolset, with different configurations of that toolset specified for use in various applications. MPEG-4 provides several toolsets that have no particular relationship to each other, each with a different target function. The Profiles of MPEG-4 Audio (subclause **1.5.2**) specify which of these tools are used together for various applications.

  Further, in previous MPEG standards, a single (perhaps multi-channel or multi-language) piece of content was transmitted. In contrast, MPEG-4 supports a much more flexible concept of a *soundtrack*. Multiple tools may be used to transmit several *audio objects*, and when using multiple tools together an *audio composition* system is used to create a single soundtrack from the several audio substreams. User interaction, terminal capability, and speaker configuration may be used when determining how to produce a single soundtrack from the component objects. This capability gives MPEG-4 significant advantages in quality and flexibility when compared to previous audio standards.

- **MPEG-4 provides capabilities for synthetic sound.** In natural sound coding, an existing sound is compressed by a server, transmitted and decompressed at the receiver. This type of coding is the subject of many existing standards for sound compression. In contrast, MPEG-4 standardizes a novel paradigm in which synthetic sound descriptions, including synthetic speech and synthetic music, are transmitted and then *synthesized* into sound at the receiver. Such capabilities open up new areas of very-low-bitrate but still very-high-quality coding.

- **MPEG-4 provides capabilities for Error Robustness.** Improved error robustness for AAC is provided by a set of error resilience tools. These tools reduce the perceived degradation of the decoded audio signal that is

caused by corrupted bits in the bitstream. Improved error robustness capabilities for all coding tools are provided through the error resilient bitstream payload syntax. This tool supports advanced channel coding techniques, which can be adapted to the special needs of given coding tools and a given communications channel. This error resilient bitstream payload syntax is mandatory for all error resillient object types.

The error protection tool (EP tool) provides unequal error protection (UEP) for MPEG-4 Audio in conjunction with the error resilient bitstream payload. UEP is an efficient method to improve the error robustness of source coding schemes. It is used by various speech and audio coding systems operating over error-prone channels such as mobile telephone networks or Digital Audio Broadcasting (DAB). The bits of the coded signal representation are first grouped into different classes according to their error sensitivity. Then error protection is individually applied to the different classes, giving better protection to more sensitive bits.

- **MPEG-4 provides capabilities for Scalability.** Previous MPEG Audio standards provided a single bitrate, single bandwidth toolset, with different configurations of that toolset specified for use in various applications. MPEG-4 provides several bitrate and bandwidth options within a single bitstream, providing a scalability functionality that permits a given bitstream to scale to the requirement of different channels and applications or to be responsive to a given channel that has dynamic throughput characteristics. The tools specified in MPEG-4 are the state-of-the-art tools providing scalable compression of speech and audio signals.

As with previous MPEG standards, MPEG-4 does not standardize methods for encoding sound. Thus, content authors are left to their own decisions as to the best method of creating bitstreams. At the present time, methods to automatically convert natural sound into synthetic or multi-object descriptions are not mature; therefore, most immediate solutions will involve interactively-authoring the content stream in some way. This process is similar to current schemes for MIDI-based and multi-channel mixdown authoring of soundtracks.

Capabilities

Overview of capabilities

The MPEG-4 Audio tools can be broadly organized into several categories:

Speech tools for the transmission and decoding of synthetic and natural speech.
Audio tools for the transmission and decoding of recorded music and other audio soundtracks.
Synthesis tools for very low bitrate description and transmission, and terminal-side synthesis, of synthetic music and other sounds.
Composition tools for object-based coding, interactive functionality, and audiovisual synchronization.
Scalability tools for the creation of bitstreams that can be transmitted, without recoding, at several different bitrates.
Upstream tools for the dynamic control the streaming of the server for bitrate control and quality feedback control.
Error robustness (including error resilience as well as error protection).

Each of these types of tools will be described in more detail in the following subclauses.

MPEG-4 speech coding tools

Two types of speech coding tools are provided in MPEG-4. The *natural* speech tools allow the compression, transmission, and decoding of human speech, for use in telephony, personal communication, and surveillance applications. The *synthetic* speech tool provides an interface to text-to-speech synthesis systems; using synthetic speech provides very-low-bitrate operation and built-in connection with facial animation for use in low-bitrate video teleconferencing applications. Each of these tools will be discussed.

Natural speech coding

The MPEG-4 speech coding toolset covers the compression and decoding of natural speech sound at bitrates ranging between 2 and 24 kbit/s. When variable bitrate coding is allowed, coding at even less than 2 kbit/s, for example an average bitrate of 1.2 kbit/s, is also supported. Two basic speech coding techniques are used: One is a parametric speech coding algorithm, HVXC (Harmonic Vector eXcitation Coding), for very low bit rates; and the other is a CELP (Code Excited Linear Prediction) coding technique. The MPEG-4 speech coders target

applications from mobile and satellite communications, to Internet telephony, to packaged media and speech databases. It meets a wide range of requirements encompassing bitrate, functionality and sound quality, and is specified in subparts 2 and 3.

MPEG-4 HVXC operates at fixed bitrates between 2.0 kbit/s and 4.0 kbit/s using a bitrate scalability technique. It also operates at lower bitrates, typically 1.2-1.7 kbit/s, using a variable bitrate technique. HVXC provides communications-quality to near-toll-quality speech in the 100-3800 Hz band at 8kHz sampling rate. HVXC also allows independent change of speed and pitch during decoding, which is a powerful functionality for fast access to speech databases. HVXC functionalities including 2.0-4.0 kbit/s fixed bitrate mode and 2.0 kbit/s maximum variable bitrate mode.

Error Resilient (ER) HVXC extends operation of the variable bitrate mode to 4.0 kbit/s to allow higher quality variable rate coding. The ER HVXC therefore provides fixed bitrate modes of 2.0 - 4.0kbit/s and a variable bitrate of either less than 2.0kbit/s or less than 4.0kbit/s, both in scalable and non-scalable modes. In the variable bitrate modes, non-speech parts are detected in unvoiced signals, and a smaller number of bits are used for these non-speech parts to reduce the average bitrate. ER HVXC provides communications-quality to near-toll-quality speech in the 100-3800 Hz band at 8kHz sampling rate. When the variable bitrate mode is allowed, operation at lower average bitrate is possible. Coded speech using variable bitrate mode at typical bitrates of 1.5kbit/s average, and at typical bitrate of 3.0kbit/s average has essentially the same quality as 2.0 kbit/s fixed rate and 4.0 kbit/s fixed rate respectively. The functionality of pitch and speed change during decoding is supported for all modes. ER HVXC has a bitstream syntax with the error sensitivity classes to be used with the EP-Tool, and the error concealment functionality is supported for use in error-prone channels such as mobile communication channels. The ER HVXC speech coder targets applications from mobile and satellite communications, to Internet telephony, to packaged media and speech databases.

MPEG-4 CELP is a well-known coding algorithm with new functionality. Conventional CELP coders offer compression at a single bit rate and are optimized for specific applications. Compression is one of the functionalities provided by MPEG-4 CELP, but MPEG-4 also enables the use of one basic coder in multiple applications. It provides scalability in bitrate and bandwidth, as well as the ability to generate bitstreams at arbitrary bitrates. The MPEG-4 CELP coder supports two sampling rates, namely, 8 and 16 kHz. The associated bandwidths are 100 – 3800 Hz for 8 kHz sampling and 50 – 7000 Hz for 16 kHz sampling. The silence compression tool comprises a Voice Activity Detector (VAD), a Discontinuous Transmission (DTX) unit and a Comfort Noise Generator (CNG) module. The tool encodes/decodes the input signal at a lower bitrate during the non-active-voice (silent) frames. During the active-voice (speech) frames, MPEG-4 CELP encoding and decoding are used.

The silence compression tool reduces the average bitrate thanks to compression at a lower-bitrate for silence. In the encoder, a voice activity detector is used to distinguish between regions with normal speech activity and those with silence or background noise. During normal speech activity, the CELP coding is used. Otherwise a Silence Insertion Descriptor (SID) is transmitted at a lower bitrate. This SID enables a Comfort Noise Generator (CNG) in the decoder. The amplitude and the spectral shape of this comfort noise are specified by energy and LPC parameters in methods similar to those used in a normal CELP frame. These parameters are optionally re-transmitted in the SID and thus can be updated as required.

MPEG has conducted extensive verification testing in realistic listening conditions in order to prove the efficacy of the speech coding toolset.

Text-to-speech interface

Text-to-speech (TTS) capability is becoming a rather common media type and plays an important role in various multi-media application areas. For instance, by using TTS functionality, multimedia content with narration can be easily created without recording natural speech. Before MPEG-4, however, there was no way for a multimedia content provider to easily give instructions to an unknown TTS system. With **MPEG-4 TTS Interface**, a single common interface for TTS systems is standardized. This interface allows speech information to be transmitted in the International Phonetic Alphabet (IPA), or in a textual (written) form of any language. It is specified in subpart 6.

The **MPEG-4 Hybrid/Multi-Level Scalable TTS Interface** is a superset of the conventional TTS framework. This extended TTS Interface can utilize prosodic information taken from natural speech in addition to input text and can thus generate much higher-quality synthetic speech. The interface and its bitstream format is scalable in terms of this added information; for example, if some parameters of prosodic information are not available, a decoder can

generate the missing parameters by rule. Normative algorithms for speech synthesis and text-to-phoneme translation are not specified in MPEG-4, but to meet the goal that underlies the MPEG-4 TTS Interface, a decoder should fully utilize all the provided information according to the user's requirements level.

As well as an interface to Text-to-speech synthesis systems, MPEG-4 specifies a joint coding method for phonemic information and facial animation (FA) parameters and other animation parameters (AP). Using this technique, a single bitstream may be used to control both the Text-to-Speech Interface and the Facial Animation visual object decoder (see ISO/IEC 14496-2 Annex C). The functionality of this extended TTS thus ranges from conventional TTS to natural speech coding and its application areas, from simple TTS to audio presentation with TTS and motion picture dubbing with TTS.

MPEG-4 general audio coding tools

MPEG-4 standardizes the coding of natural audio at bitrates ranging from 6 kbit/s up to several hundred kbit/s per audio channel for mono, two-channel-, and multi-channel-stereo signals. General high-quality compression is provided by incorporating the MPEG-2 AAC standard (ISO/IEC 13818-7), with certain improvements, as **MPEG-4 AAC**. At 64 kbit/s/channel and higher ranges, this coder has been found in verification testing under rigorous conditions to meet the criterion of "indistinguishable quality" as defined by the European Broadcasting Union.

Subpart 4 of MPEG-4 specifies the AAC tool set, **MPEG-4 Twin-VQ** and BSAC, in the General Audio (GA) coder. This coding technique uses a perceptual filterbank, a sophisticated masking model, noise-shaping techniques, channel coupling, and noiseless coding and bit-allocation to provide the maximum compression within the constraints of providing the highest possible quality. Psychoacoustic coding standards developed by MPEG have represented the state-of-the-art in this technology since MPEG-1 Audio; MPEG-4 General Audio coding continues this tradition.

For bitrates ranging from 6 kbit/s to 64 kbit/s per channel, the MPEG-4 standard provides extensions to the GA coding tools, AAC, TwinVQ, and BSAC, that allow the content author to achieve the highest quality coding at the desired bitrate. Furthermore, various bit rate scalability options are available within the GA coder. The low-bitrate techniques and scalability modes provided within this tool set have also been verified in formal tests by MPEG.

The **MPEG-4 low delay** coding functionality provides the abillty to extend the usage of generic low bitrate audio coding to applications requiring a very low delay in the encoding / decoding chain (e.g. full-duplex real-time communications). In contrast to traditional low delay coders based on speech coding technology, the concept of this low delay coder is based on general perceptual audio coding and is thus suitable for a wide range of audio signals. Specifically, it is derived from the proven architecture of MPEG-2/4 Advanced Audio Coding (AAC) and all capabilities for coding of 2 (stereo) or more sound channels (multi-channel) are available within the low delay coder. It operates at up to 48 kHz sampling rate and uses a frame length of 512 or 480 samples, compared to the 1024 or 960 samples used in standard MPEG-2/4 AAC to enable coding of general audio signals with an algorithmic delay not exceeding 20 ms. Also the size of the window used in the analysis and synthesis filterbank is reduced by a factor of 2. No block switching is used to avoid the "look-ahead'" delay due to the block switching decision. To reduce pre-echo artefacts in the case of transient signals, window shape switching is provided instead. For non-transient portions of the signal a sine window is used, while a so-called low overlap window is used for transient portions. Use of the bit reservoir is minimized in the encoder in order to reach the desired target delay. As one extreme case, no bit reservoir is used at all.

The **MPEG-4 BSAC** is used in combination with the AAC coding tools and replaces the noiseless coding of the quantized spectral data and the scalefactors. The MPEG-4 BSAC provides fine grain scalability in steps of 1 kbit/s per audio channel, i.e. 2 kbit/s steps for a stereo signal. One base layer bitstream and many small enhancement layer bitstreams are used. To obtain fine step scalability, a bit-slicing scheme is applied to the quantized spectral data. First the quantized spectral values are grouped into frequency bands. Each of these groups contains the quantized spectral values in their binary representation. Then the bits of a group are processed in slices according to their significance. Thus all most significant bits (MSB) of the quantized values in a group are processed first. These bit-slices are then encoded using an arithmetic coding scheme to obtain entropy coding with minimal redundancy. In order to implement fine grain scalability efficiently using MPEG-4 Systems tools, the fine grain audio data can be grouped into large-step layers and these large-step layers further grouped by concatenating large-step layers from several sub-frames. Furthermore, the configuration of the payload transmitted over an Elementary Stream (ES) can be changed dynamically (by means of the MPEG-4 backchannel capability) depending on the environment, such as network traffic or user interaction. This means that BSAC can allow for real-time adjustments to the quality of service. In addition to fine grain scalablity, it can improve the quality of an

audio signal that is decoded from a bitstream transmitted over an error-prone channel, such as a mobile communication networks or Digital Audio Broadcasting (DAB) channel.

Subpart 7 of MPEG-4 specifies the parametric audio coding tool **MPEG-4 HILN** (Harmonic and Individual Lines plus Noise) to code non-speech signals like music at bitrates of 4 kbit/s and higher using a parametric representation of the audio signal. The basic idea of this technique is to decompose the input signal into audio objects which are described by appropriate source models and represented by model parameters. Object models for sinusoids, harmonic tones, and noise are utilized in the HILN coder. HILN allows independent change of speed and pitch during decoding. Furthermore HILN can be combined with MPEG-4 parametric speech coding (HVXC) to form an integrated parametric coder covering a wider range of signals and bitrates.

The Parametric Audio Coding tools combine very low bitrate coding of general audio signals with the possibility of modifying the playback speed or pitch during decoding without the need for an effects processing unit. In combination with the speech and audio coding tools in MPEG-4, improved overall coding efficiency is expected for applications of object based coding allowing selection and/or switching between different coding techniques.

This approach allows one to introduce a more advanced source model than just assuming a stationary signal for the duration of a frame, which motivates the spectral decomposition used in e.g. the MPEG-4 General Audio Coder. As known from speech coding, where specialized source models based on the speech generation process in the human vocal tract are applied, advanced source models can be advantageous, especially for very low bitrate coding schemes.

Due to the very low target bitrates, only the parameters for a small number of objects can be transmitted. Therefore a perception model is employed to select those objects that are most important for the perceptual quality of the signal.

In HILN, the frequency and amplitude parameters are quantized according to the “just noticeable differences” known from psychoacoustics. The spectral envelope of the noise and the harmonic tones are described using LPC modeling as known from speech coding. Correlation between the parameters of one frame and those of consecutive frames is exploited by parameter prediction. Finally, the quantized parameters are entropy coded and multiplexed to form a bitstream.

A very interesting property of this parametric coding scheme arises from the fact that the signal is described in terms of frequency and amplitude parameters. This signal representation permits speed and pitch change functionality by simple parameter modification in the decoder. The HILN Parametric Audio Coder can be combined with MPEG-4 Parametric Speech Coder (HVXC) to form an integrated parametric coder covering a wider range of signals and bitrates. This integrated coder supports speed and pitch change. Using a speech/music classification tool in the encoder, it is possible to automatically select the HVXC for speech signals and the HILN for music signals. Such automatic HVXC/HILN switching was successfully demonstrated and the classification tool is described in the informative Annex of the MPEG-4 standard.

MPEG-4 Audio synthesis tools

The MPEG-4 toolset providing general audio synthesis capability is called **MPEG-4 Structured Audio**, and it is described in subpart 5 of ISO/IEC 14496-3. MPEG-4 Structured Audio (the SA coder) provides very general capabilities for the description of synthetic sound, and the normative creation of synthetic sound in the decoding terminal. High-quality stereo sound can be transmitted at bitrates from 0 kbit/s (no continuous cost) to 2-3 kbit/s for extremely expressive sound using these tools.

Rather than specify a particular method of synthesis, SA specifies a flexible language for describing methods of synthesis. This technique allows content authors two advantages. First, the set of synthesis techniques available is not limited to those that were envisioned as useful by the creators of the standard; any current or future method of synthesis may be used in MPEG-4 Structured Audio. Second, the creation of synthetic sound from structured descriptions is normative in MPEG-4, so sound created with the SA coder will sound the same on any terminal.

Synthetic audio is transmitted via a set of *instrument* modules that can create audio signals under the control of a *score*. An instrument is a small network of signal-processing primitives that control the parametric generation of sound according to some algorithm. Several different instruments may be transmitted and used in a single Structured Audio bitstream. A score is a time-sequenced set of commands that invokes various instruments at specific times to contribute their output to an overall music performance. The format for the description of

instruments—SAOL, the Structured Audio Orchestra Language—and that for the description of scores—SASL, the Structured Audio Score Language—are specified in subpart 6.

Efficient transmission of sound samples, also called *wavetables*, for use in sampling synthesis is accomplished by providing interoperability with the MIDI Manufacturers Association Downloaded Sounds Level 2 (DLS-2) standard, which is normatively referenced by the Structured Audio standard. By using the DLS-2 format, the simple and popular technique of wavetable synthesis can be used in MPEG-4 Structured Audio soundtracks, either by itself or in conjunction with other kinds of synthesis using the more general-purpose tools. To further enable interoperability with existing content and authoring tools, the popular MIDI (Musical Instrument Digital Interface) control format can be used instead of, or in addition to, scores in SASL for controlling synthesis.

Through the inclusion of compatibility with MIDI standards, MPEG-4 Structured Audio thus represents a unification of the current technique for synthetic sound description (MIDI-based wavetable synthesis) with that of the future (general-purpose algorithmic synthesis). The resulting standard solves problems not only in very-low-bitrate coding, but also in virtual environments, video games, interactive music, karaoke systems, and many other applications.

MPEG-4 Audio composition tools

The tools for audio composition, like those for visual composition, are specified in the MPEG-4 Systems standard (ISO/IEC 14496-1). However, since readers interested in audio functionality are likely to look here first, a brief overview is provided.

Audio composition is the use of multiple individual “audio objects” and mixing techniques to create a single soundtrack. It is analogous to the process of recording a soundtrack in a multichannel mix, with each musical instrument, voice actor, and sound effect on its own channel, and then “mixing down” the multiple channels to a single channel or single stereo pair. In MPEG-4, the multichannel mix itself may be transmitted, with each audio source using a different coding tool, and a set of instructions for mixdown also transmitted in the bitstream. As the multiple audio objects are received, they are decoded separately, but not played back to the listener; rather, the instructions for mixdown are used to prepare a single soundtrack from the “raw material” given in the objects. This final soundtrack is then played for the listener.

An example serves to illustrate the efficacy of this approach. Suppose, for a certain application, we wish to transmit the sound of a person speaking in a reverberant environment over stereo background music, at very high quality. A traditional approach to coding would demand the use of a general audio coding at 32 kbit/s/channel or above; the sound source is too complex to be well-modeled by a simple model-based coder. However, in MPEG-4 we can represent the soundtrack as the conjunction of several objects: a speaking person passed through a reverberator added to a synthetic music track. We transmit the speaker’s voice using the CELP tool at 16 kbit/s, the synthetic music using the SA tool at 2 kbit/s, and allow a small amount of overhead (only a few hundreds of bytes as a fixed cost) to describe the stereo mixdown and the reverberation. Using MPEG-4 and an object-based approach thus allows us to describe in less than 20 kbit/s total a bitstream that might require 64 kbit/s to transmit with traditional coding, at equivalent quality.

Additionally, having such structured soundtrack information present in the decoding terminal allows more sophisticated client-side interaction to be included. For example, the listener can be allowed (if the content author desires) to request that the background music be muted. This functionality would not be possible if the music and speech were coded into the same audio track.

With the **MPEG-4 Binary Format for Scenes (BIFS),** specified in MPEG-4 Systems, a subset tool called AudioBIFS allows content authors to describe sound scenes using this object-based framework. Multiple sources may be mixed and combined, and interactive control provided for their combination. Sample-resolution control over mixing is provided in this method. Dynamic download of custom signal-processing routines allows the content author to exactly request a particular, normative, digital filter, reverberator, or other effects-processing routine. Finally, an interface to terminal-dependent methods of 3-D audio spatialisation is provided for the description of virtual-reality and other 3-D sound material.

As AudioBIFS is part of the general BIFS specification, the same framework is used to synchronize audio and video, audio and computer graphics, or audio with other material. Please refer to ISO/IEC 14496-1 (MPEG-4 Systems) for more information on AudioBIFS and other topics in audiovisual synchronization.

MPEG-4 Audio scalability tools

Many of the bitstream types in MPEG-4 are *scalable* in one manner or another. Several types of scalability in the standard are discussed below.

Bitrate scalability allows a bitstream to be parsed into a bitstream of lower bitrate such that the combination can still be decoded into a meaningful signal. The bitstream parsing can occur either during transmission or in the decoder. Scalability is available within each of the natural audio coding schemes, or by a combination of different natural audio coding schemes.

Bandwidth scalability is a particular case of bitrate scalability, whereby part of a bitstream representing a part of the frequency spectrum can be discarded during transmission or decoding. This is available for the CELP speech coder, where an extension layer converts the narrow band base layer encoder into a wide band speech coder. Also the general audio coding tools which all operate in the frequency domain offer a very flexible bandwidth control for the different coding layers.

Encoder complexity scalability allows encoders of different complexity to generate valid and meaningful bitstreams. An example for this is the availability of a high quality and a low complexity excitation module for the wideband CELP coder allowing to choose between significant lower encoder complexity or optimized coding quality.

Decoder complexity scalability allows a given bitstream to be decoded by decoders of different levels of complexity. A subtype of decoder complexity scalability is *graceful degradation*, in which a decoder dynamically monitors the resources available, and scales down the decoding complexity (and thus the audio quality) when resources are limited. The Structured Audio decoder allows this type of scalability; a content author may provide (for example) several different algorithms for the synthesis of piano sounds, and the content itself decides, depending on available resources, which one to use.

MPEG-4 Audio Upstream

The **MPEG-4 upstream** or backchannel allows a user on a remote side to dynamically control the streaming MPEG-4 content from a server. Backchannel streams carrying the user interaction information.

MPEG-4 Audio Error robustness

The error robustness tools provide improved performance on error-prone transmission channels. They are comprised of codec specific error resilience tools and an common error protection tool.

Error resilience tools for AAC

Several tools are provided to increase the error resilience for AAC. These tools improve the perceived audio quality of the decoded audio signal in case of corrupted bitstreams, which may occur e. g. in the presence of noisy transmission channels.

- The *Virtual CodeBooks tool (VCB11)* extends the sectioning information of an AAC bitstream. This permits the detection of serious errors within the spectral data of an MPEG-4 AAC bitstream. Virtual codebooks are used to limit the largest absolute value possible within a any scalefactor band that uses escape values. While using to the same codebook used by codebook 11, the sixteen virtual codebooks introduced by VCB11 provide sixteen different limitations of the spectral values belonging to the corresponding subclause. Therefore, errors in the transmission of spectral data that result in spectral values exceeding the indicated limit can be located and appropriately concealed.

- The *Reversible Variable Length Coding tool (RVLC)* replaces the Huffman and DPCM coding of the scalefactors in an AAC bitstream. The RVLC uses symmetric codewords to enable both forward and backward decoding of the scalefactor data. In order to have a starting point for backward decoding, the total number of bits of the RVLC part of the bitstream is transmitted. Because of the DPCM coding of the scalefactors, also the value of the last scalefactor is transmitted to enable backward DPCM decoding. Since not all nodes of the RVLC code tree are used as codewords, some error detection is also possible.

- The *Huffman codeword reordering (HCR)* algorithm for AAC spectral data is based on the fact that some of the codewords can be placed at known positions so that these codewords can be decoded independent of any error within other codewords. Therefore, this algorithm avoids error propagation to those codewords,

the so-called priority codewords (PCW). To achieve this, segments of known length are defined and those codewords are placed at the beginning of these segments. The remaining codewords (non-priority codewords, non-PCW) are filled into the gaps left by the PCWs using a special algorithm that minimizes error propagation to the non-PCWs codewords. This reordering algorithm does not increase the size of spectral data. Before applying the reordering algorithm, the PCWs are determined by sorting the codewords according to their importance.

Error protection

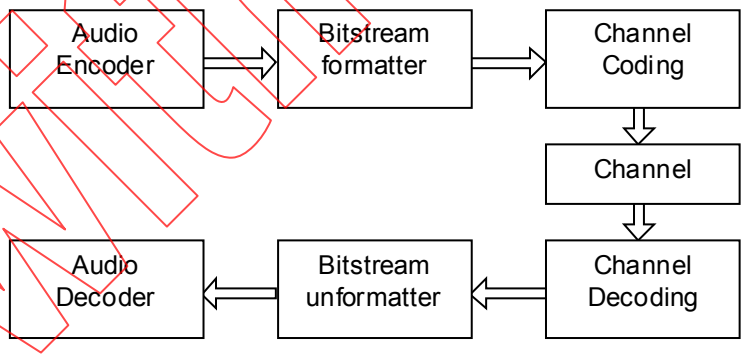
The EP tool provides unequal error protection. It receives several classes of bits from the audio coding tools, and then applies forward error correction codes (FEC) and/or cyclic redundancy codes (CRC) for each class, according to its error sensitivity.

The error protection tool (EP tool) provides the unequal error protection (UEP) capability to the set of ISO/IEC 14496-3 codecs. Main features of this tool are:

- *providing a set of error correcting/detecting codes with wide and small-step scalability, both in performance and in redundancy*
- *providing a generic and bandwidth-efficient error protection framework, which covers both fixed-length frame bitstreams and variable-length frame bitstreams*
- *providing a UEP configuration control with low overhead.*

Error resilient bitstream reordering

Error resilient bitstream reordering allows the effective use of advanced channel coding techniques like unequal error protection (UEP), which can be perfectly adapted to the needs of the different coding tools. The basic idea is to rearrange the audio frame content depending on its error sensitivity in one or more instances belonging to different error sensitivity categories (ESC). This rearrangement can be either data element-wise or even bit-wise. An error resilient bitstream frame is build by concatenating these instances.



The basic principle is depicted in the figure above. A bitstream is reordered according to the error sensitivity of single bitstream elements or even single bits. This new arranged bitstream is channel coded, transmitted and channel decoded. Prior to audio decoding, the bitstream is rearranged to its original order. The reordered syntax, that is the syntax of the bitstream transmitted over the channel, is defined in the corresponding subparts.

Information technology — Coding of audio-visual objects —

Part 3:
Audio

Structure of this part of ISO/IEC 14496:

This part of ISO/IEC 14496 contains seven subparts:

Subpart 1: Main

Subpart 2: Speech coding — HVXC

Subpart 3: Speech coding — CELP

Subpart 4: General Audio coding (GA) — AAC, TwinVQ, BSAC

Subpart 5: Structured Audio (SA)

Subpart 6: Text To Speech Interface (TTSI)

Subpart 7: Parametric Audio Coding — HILN

Contents for Subpart 1

Subpart 1: Main

1.1 Scope

ISO/IEC 14496-3 (MPEG-4 Audio) is a new kind of audio standard that integrates many different types of audio coding: natural sound with synthetic sound, low bitrate delivery with high-quality delivery, speech with music, complex soundtracks with simple ones, and traditional content with interactive and virtual-reality content. By standardizing individually sophisticated coding tools as well as a novel, flexible framework for audio synchronization, mixing, and downloaded post-production, the developers of the MPEG-4 Audio standard have created new technology for a new, interactive world of digital audio.

MPEG-4, unlike previous audio standards created by ISO/IEC and other groups, does not target a single application such as real-time telephony or high-quality audio compression. Rather, MPEG-4 Audio is a standard that applies to *every* application requiring the use of advanced sound compression, synthesis, manipulation, or playback. The subparts that follow specify the state-of-the-art coding tools in several domains; however, MPEG-4 Audio is more than just the sum of its parts. As the tools described here are integrated with the rest of the MPEG-4 standard, exciting new possibilities for object-based audio coding, interactive presentation, dynamic soundtracks, and other sorts of new media, are enabled.

Since a single set of tools is used to cover the needs of a broad range of applications, *interoperability* is a natural feature of systems that depend on the MPEG-4 Audio standard. A system that uses a particular coder—for example, a real-time voice communication system making use of the MPEG-4 speech coding toolset—can easily share data and development tools with other systems, even in different domains, that use the same tool—for example, a voicemail indexing and retrieval system making use of MPEG-4 speech coding. A multimedia terminal that can decode the Natural Audio Profile of MPEG-4 Audio has audio capabilities that cover the entire spectrum of audio functionality available today and into the future.

1.2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 14496. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 14496 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 11172-3:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio*

ITU-T Rec. H.222.0 (1995) | ISO/IEC 13818-1:2000, *Information technology – Generic coding of moving pictures and associated audio information: Systems*

ISO/IEC 13818-3:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio*

ISO/IEC 13818-7:1997, *Information technology – Generic coding of moving pictures and associated audio information – Part 7: Advanced Audio Coding (AAC)*

ITU-T Rec. H.223 (2001), *Multiplexing protocol for low bit rate multimedia communication*

MIDI Manufacturers Association, 1996, *The Complete MIDI 1.0 Detailed Specification* v. 96.2

MIDI Manufacturers Association, 1998, *The MIDI Downloadable Sounds Specification* v. 98.2

Contents for Subpart 2

Subpart 2: Speech coding - HVXC

2.1 Scope

MPEG-4 parametric speech coding uses Harmonic Vector eXcitation Coding (HVXC) algorithm, where harmonic coding of LPC residual signals for voiced segments and Vector eXcitation Coding (VXC) for unvoiced segments are employed. HVXC allows coding of speech signals at 2.0 kbps and 4.0 kbps with a scalable scheme, where 2.0 kbps decoding is possible not only using the 2.0 kbps bit-stream but also using a 4.0 kbps bit-stream. HVXC also provides variable bit rate coding where a typical average bit-rate is around 1.2-1.7 kbit/s. Independent change of speed and pitch during decoding is possible, which is a powerful functionality for fast data base search. The frame length is 20 ms, and one of four different algorithmic delays, 33.5 ms, 36ms, 53.5 ms, 56 ms can be selected.

Furthermore as an extension of HVXC, ER_HVXC object type offers error resilient syntax and the 4.0kbit/s variable bitrate mode.

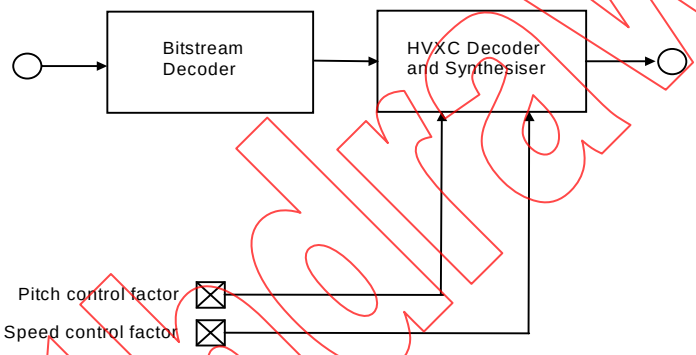


Figure 2.1 - Block diagram of the parametric speech decoder

Contents for Subpart 3

Subpart 3: Speech Coding - CELP

3.1 Scope

3.1.1 General description of the CELP decoder

This subclause provides a brief overview of the CELP (Code Excited Linear Prediction) decoder. A basic block diagram of the CELP decoder is given in Figure 3.1.

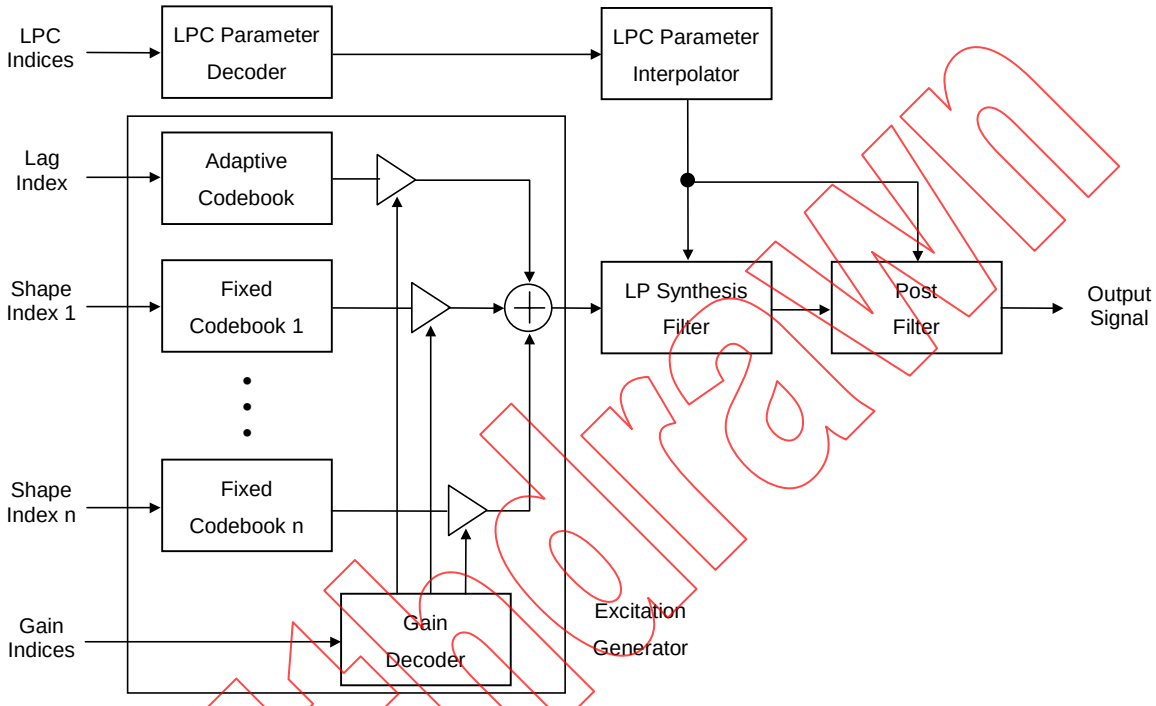


Figure 3.1 - Block diagram of a CELP decoder

The CELP decoder primarily consists of an excitation generator and a synthesis filter. Additionally, CELP decoders often include a post-filter. The excitation generator has an adaptive codebook to model periodic components, fixed codebooks to model random components and a gain decoder to represent a speech signal level. Indices for the codebooks and gains are provided by the encoder. The codebook indices (pitch-lag index for the adaptive codebook and shape index for the fixed codebook) and gain indices (adaptive and fixed codebook gains) are used to generate the excitation signal. It is then filtered by the linear predictive synthesis filter (LP synthesis filter). Filter coefficients are reconstructed using the LPC indices, then are interpolated with the filter coefficients of successive analysis frames. Finally, a post-filter can optionally be applied in order to enhance the speech quality.

Contents for Subpart 4

Subpart 4: General Audio Coding (GA) – AAC, TwinVQ, BSAC

4.1 Scope

The General Audio (GA) coding subpart of MPEG-4 Audio is mainly intended to be used for generic audio coding at all but the lowest bitrates. Typically, GA encoding is used for complex music material in mono from 6 kbit/s per channel and for stereo signals from 12 kbit/s per stereo signal up to broadcast quality audio at 64 kbit/s or more per channel. MPEG-4 coded material can be represented either by a single set of data, like in MPEG-1 and MPEG-2 Audio, or by several subsets which allow the decoding at different quality levels, depending on the number of subsets being available at the decoder side (bitrate scalability).

MPEG-2 Advanced Audio Coding (AAC) syntax (including support for multi-channel audio) is fully supported by MPEG-4 Audio GA coding. All the features and possibilities of the MPEG-2 AAC standard also apply to MPEG-4. AAC has been tested to allow for ITU-R 'indistinguishable' quality according to [4] at data rates of 320 kb/s for five full-bandwidth channel audio signals. In MPEG-4 the tools derived from MPEG-2 AAC are available together with other MPEG-4 GA coding tools which provide additional functionalities, like bit rate scalability and improved coding efficiency at very low bit rates. Bit rate scalability is either achieved with only GA coding tools, or by using a combination with an external (non-GA, e.g. CELP) core coder.

MPEG-4 GA coding is not restricted to some fixed bitrates but supports a wide range of bitrates and variable rate coding. While efficient mono, stereo and multi-channel coding is possible using extended, MPEG-2 AAC derived tools, the document also provides extensions to this tool set which allow mono/stereo scalability, where a mono signal can be extracted by decoding only subsets of the encoded stereo stream.

4.1.1 *Technical Overview*

4.1.1.1 Encoder and Decoder Block Diagrams

The block diagrams of the GA encoder and decoder reflect the structure of MPEG-4 GA coding. In general, there are the MPEG-2 AAC related tools with MPEG-4 add-ons for some of them and the tools related to the TwinVQ quantization and coding. The TwinVQ is an alternative module for the AAC-type quantization and it is based on an interleaved vector quantization and LPC (Linear Predictive Coding) spectral estimation. It operates from 6 kbit/s/ch and is recommended to be used below 16 kbit/s/ch with constant bitrate.

The basic structure of the MPEG-4 GA system is shown in Figure 4.1 and Figure 4.2. The data flow in this diagram is from left to right, top to bottom. The functions of the decoder are to find the description of the quantized audio spectra in the bitstream, decode the quantized values and other reconstruction information, reconstruct the quantized spectra, process the reconstructed spectra through whatever tools are active in the bitstream in order to arrive at the actual signal spectra as described by the input bitstream, and finally convert the frequency domain spectra to the time domain, with or without an optional gain control tool. Following the initial reconstruction and scaling of the spectrum reconstruction, there are many optional tools that modify one or more of the spectra in order to provide more efficient coding. For each of the optional tools that operate in the spectral domain, the option to "pass through" is retained, and in all cases where a spectral operation is omitted, the spectra at its input are passed directly through the tool without modification.

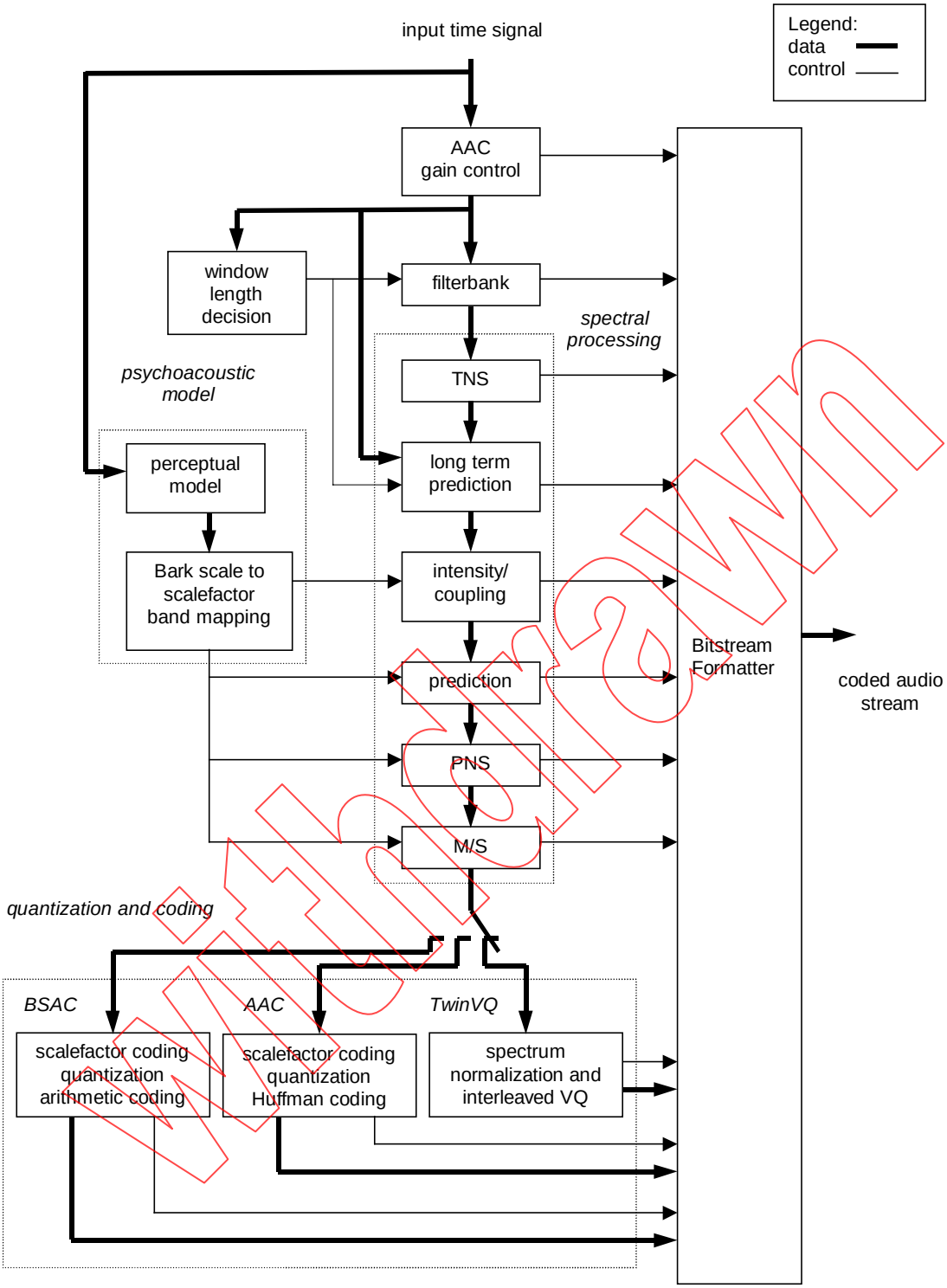


Figure 4.1 – Block diagram GA non scalable encoder

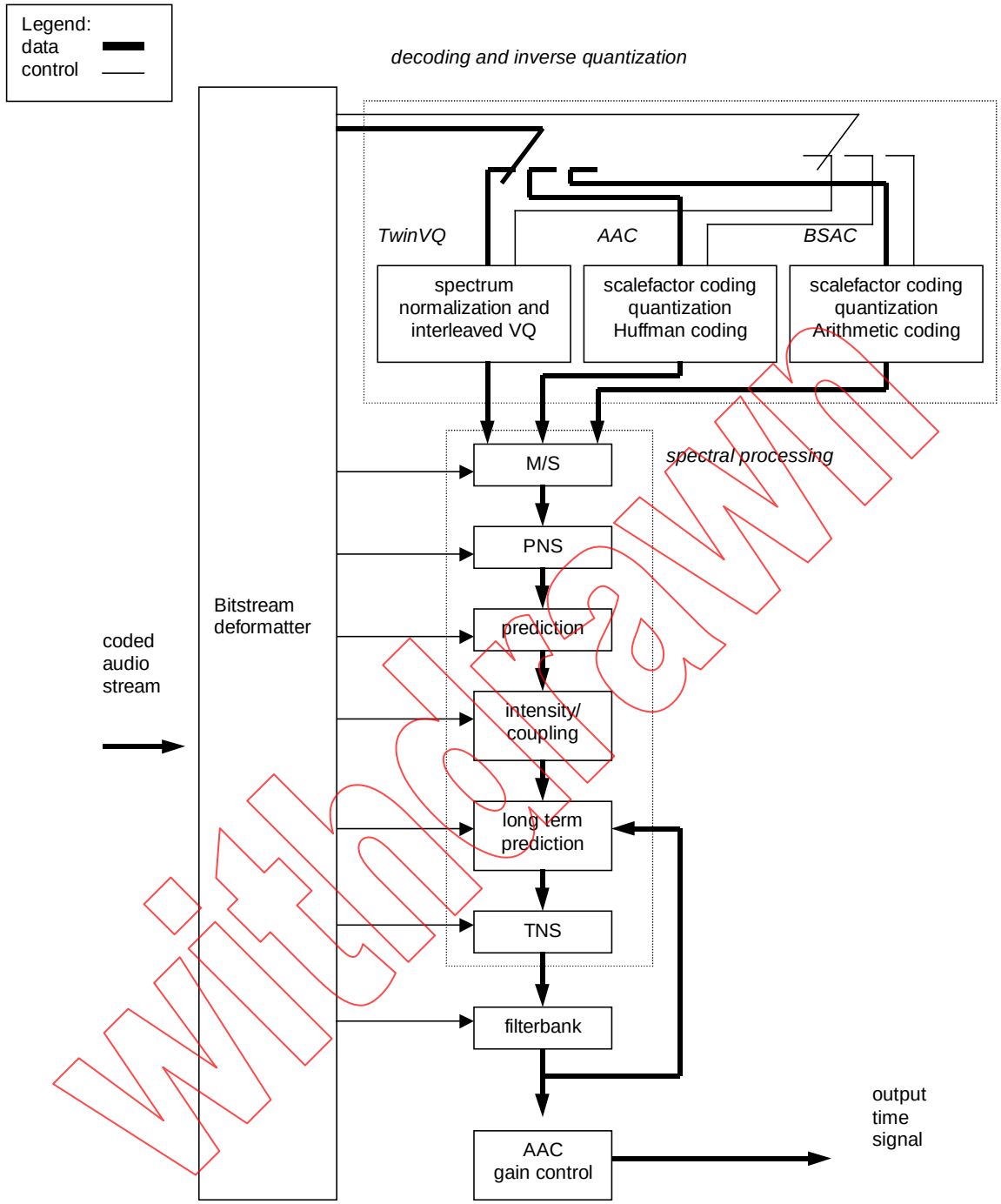


Figure 4.2 – Block diagram of the GA non scalable decoder

4.1.1.2 Overview of the Encoder and Decoder Tools

The input to the bitstream demultiplexer tool is the MPEG-4 GA bitstream. The demultiplexer separates the bitstream into the parts for each tool, and provides each of the tools with the bitstream information related to that tool.

The outputs from the bitstream demultiplexer tool are:

- The quantized (and optionally noiselessly coded) spectra represented by either

- the sectioning information and the noiselessly coded spectra (AAC) or
- a set of indices of code vectors (TwinVQ) or
- the arithmetic model information and the noiselessly coded spectra (BSAC)

- The M/S decision information (optional)
- The predictor side information (optional)
- The perceptual noise substitution (PNS) information (optional)
- The intensity stereo control information and coupling channel control information (both optional)
- The temporal noise shaping (TNS) information (optional)
- The filterbank control information
- The gain control information (optional)
- Bitrate scalability related side information (optional)

The AAC noiseless decoding tool takes information from the bitstream demultiplexer, parses that information, decodes the Huffman coded data, and reconstructs the quantized spectra and the Huffman and DPCM coded scalefactors.
The inputs to the noiseless decoding tool are:

- The sectioning information for the noiselessly coded spectra
- The noiselessly coded spectra

The outputs of the noiseless decoding tool are:

- The decoded integer representation of the scalefactors:
- The quantized values for the spectra

The inverse quantizer tool takes the quantized values for the spectra, and converts the integer values to the non-scaled, reconstructed spectra. This quantizer is a non-uniform quantizer.
The input to the Inverse Quantizer tool is:

- The quantized values for the spectra

The output of the inverse quantizer tool is:

- The un-scaled, inversely quantized spectra

The scalefactor tool converts the integer representation of the scalefactors to the actual values, and multiplies the un-scaled inversely quantized spectra by the relevant scalefactors.
The inputs to the scalefactors tool are:

- The decoded integer representation of the scalefactors
- The un-scaled, inversely quantized spectra

The output from the scalefactors tool is:

- The scaled, inversely quantized spectra

The M/S tool converts spectra pairs from Mid/Side to Left/Right under control of the M/S decision information, improving stereo imaging quality and sometimes providing coding efficiency.
The inputs to the M/S tool are:

- The M/S decision information
- The scaled, inversely quantized spectra related to pairs of channels

The output from the M/S tool is:

- The scaled, inversely quantized spectra related to pairs of channels, after M/S decoding

Note: The scaled, inversely quantized spectra of individually coded channels are not processed by the M/S block, rather they are passed directly through the block without modification. If the M/S block is not active, all spectra are passed through this block unmodified.

The prediction tool reverses the prediction process carried out at the encoder. This prediction process re-inserts the redundancy that was extracted by the prediction tool at the encoder, under the control of the predictor state information. This tool is implemented as a second order backward adaptive predictor.
The inputs to the prediction tool are:

- The predictor state information

- The predictor side information
- The scaled, inversely quantized spectra

The output from the prediction tool is:

- The scaled, inversely quantized spectra, after prediction is applied.

Note: If the prediction is disabled, the scaled, inversely quantized spectra are passed directly through the block without modification.

Alternatively, there is a forward adaptive long term prediction tool provided.
The inputs to the long term prediction tool are:

- The reconstructed time domain output of the decoder
- The scaled, inversely quantized spectra

The output from the long term prediction tool is:

- The scaled, inversely quantized spectra, after prediction is applied.

Note: If the prediction is disabled, the scaled, inversely quantized spectra are passed directly through the block without modification.

The perceptual noise substitution (PNS) tool implements noise substitution decoding on channel spectra by providing an efficient representation for noise-like signal components.
The inputs to the perceptual noise substitution tool are:

- The inversely quantized spectra
- The perceptual noise substitution control information

The output from the perceptual noise substitution tool is:

- The inversely quantized spectra

Note: If either part of this block is disabled, the scaled, inversely quantized spectra are passed directly through this part without modification. If the perceptual noise substitution block is not active, all spectra are passed through this block unmodified.

The intensity stereo / coupling tool implements intensity stereo decoding on pairs of spectra. In addition, it adds the relevant data from a dependently switched coupling channel to the spectra at this point, as directed by the coupling control information.
The inputs to the intensity stereo / coupling tool are:

- The inversely quantized spectra
- The intensity stereo control information and coupling control information

The output from the intensity stereo / coupling tool is:

- The inversely quantized spectra after intensity and coupling channel decoding.

Note: If either part of this block is disabled, the scaled, inversely quantized spectra are passed directly through this part without modification. The intensity stereo tool and M/S tools are arranged so that the operation of M/S and Intensity stereo are mutually exclusive on any given scalefactor band and group of one pair of spectra.

The temporal noise shaping (TNS) tool implements a control of the fine time structure of the coding noise. In the encoder, the TNS process has flattened the temporal envelope of the signal to which it has been applied. In the decoder, the inverse process is used to restore the actual temporal envelope(s), under control of the TNS information. This is done by applying a filtering process to parts of the spectral data.
The inputs to the TNS tool are:

- The inversely quantized spectra
- The TNS information

The output from the TNS block is:

- The inversely quantized spectra

Note: If this block is disabled, the inversely quantized spectra are passed through without modification.

The filterbank tool applies the inverse of the frequency mapping that was carried out in the encoder, as indicated by the filterbank control information and the presence or absence of gain control information. An inverse modified discrete cosine transform (IMDCT) is used for the filterbank tool. If the gain control tool is not used, the IMDCT

input consists of either 1024 or 128 (depending on **window_sequence** spectral coefficients (if **frameLengthFlag** is set to '0'), or of 960 or 120 spectral coefficients (if **frameLengthFlag** is set to '1'), respectively. If the gain control tool is used, the filterbank tool is configured to use four sets of either 256 or 32 coefficients, depending of the value of window_sequence.
The inputs to the filterbank tool are:

- The inversely quantized spectra
- The filterbank control information

The output(s) from the filterbank tool is (are):

- The time domain reconstructed audio signal(s).

Two alternative, but very similar versions of this tool are available. The version with a frame length of 960 samples, which is not available in ISO/IEC 13818-7, allows for an integer frame length. For example at 48 kHz sampling rate, the frame length is exactly 20 ms with this version. This is especially useful for the CELP/AAC bitrate scalability combinations, where this allows the construction of combined CELP layer frames, which have a length of a multiple of 10 ms, and AAC enhancement layer frames. However, this feature can be used for configurations with only AAC or TwinVQ coding as well.

When present, the gain control tool applies a separate time domain gain control to each of 4 frequency bands that have been created by the gain control PQF filterbank in the encoder. Then, it assembles the 4 frequency bands and reconstructs the time waveform through the gain control tool's filterbank.
The inputs to the gain control tool are:

- The time domain reconstructed audio signal(s)
- The gain control information

The output(s) from the gain control tool is (are):

- The time domain reconstructed audio signal(s)

If the gain control tool is not active, the time domain reconstructed audio signal(s) are passed directly from the filterbank tool to the output of the decoder. This tool is used for the scalable sampling rate (SSR) audio object type only.

The spectrum normalization tool converts the reconstructed flat spectra to the actual values at the decoder. The spectral envelope is specified by LPC coefficients, a Bark scale envelope, periodic peak components, and gain.
The input to the spectral normalization tool are

- The reconstructed flat spectra
- The information of LPC coefficients, a Bark scale envelope, periodic peak components and gain

The output from the spectral normalization tool is

- The reconstructed actual spectra

The interleaved VQ tool converts the vector index to the flattened spectra at the TwinVQ decoder by means of table look-up of the codebook and inverse interleaving of the spectra. Quantization noise is minimized by a weighted distortion measure at the encoder instead of an adaptive bit allocation. This is an alternative to the AAC quantization tool.
The input to the interleaved VQ tool is:

- A set of indices of the code vector.

The output from the TwinVQ tool is:

- The reconstructed flattened spectra

The Frequency Selective Switch (FSS) tool is used to control the combination of the AAC coding layer with both, TwinVQ, and CELP coding layer, if these are used as base layer coder in scalable configurations. In a second function this tool is applied to control the combination of mono and stereo coding layer in scalable configurations where both mono, and stereo coding layer are used to code a stereo input signal.

The Up-sampling Filter tool adapts the sampling rate of a CELP core coder, which can be used as base layer coder in scalable configurations, to the sampling rate of the AAC extension layer.
The input to the Upsampling Filter tool is:

- The output of a CELP core coder running at a lower sampling rate than the AAC extension layer

The output from the Up-sampling Filter tool is:

- The up-sampled CELP core coder output, matching the sampling rate of the AAC extension layer, transformed into the frequency domain with exactly the same frequency and time resolution as the AAC extension layer.

The BSAC noiseless decoding tool takes information from the bitstream demultiplexer, parses that information, decodes the Arithmetic coded data, and reconstructs the quantized spectra and the Arithmetic coded scalefactors. The BSAC noiseless coding module is an alternative to the AAC coding module. The BSAC noiseless coding is used to make the bitstream scalable and error resilient and further reduce the redundancy of the scalefactors and the quantized spectrum.
The inputs to the BSAC decoding tool are

- The Arithmetic model information for the noiselessly coded spectra
- The noiselessly coded bit-sliced data

The outputs from the BSAC decoding tool are

- The decoded integer representation of the scalefactors
- The quantized value for the spectra

The virtual codebooks (VCB11) tool can extend the part of the bitstream demultiplexer that decodes the sectioning information. The VCB11 tool gives the opportunity to detect serious errors within the spectral data of an MPEG-4 AAC bitstream.
The input of the VCB11 tool is:

- The encoded section data using virtual codebooks

The output of the VCB11 tool is:

- The decoded sectioning information

The reversible variable length coding (RVLC) tool can replace the part of the noiseless coding tool that decodes the Huffman and DPCM coded scalefactors. The RVLC tool is used to increase the error resilience for the scalefactor data within an MPEG-4 AAC bitstream.
The input of the RVLC tool is:

- The noiselessly coded scalefactors using RVLC

The output of the RVLC tool is:

- The decoded integer representation of the scalefactors

The Huffman codeword reordering (HCR) tool can extend the part of the noiseless coding tool that decodes the Huffman coded spectral data. The HCR tool is used to increase the error resilience for the spectral data within an MPEG-4 AAC bitstream.
The input of the HCR tool is:

- The sectioning information for the noiselessly coded spectra
- The noiselessly coded spectral data in an error resilient reordered manner
- The length of the longest codeword within spectral_data
- The length of spectral_data

The output of the HCR tool is:

- The quantized value of the spectra

4.2 Normative references

ISO/IEC 11172-3:1993, *Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 3: Audio*

ITU-T Rec.H.222.0 (1995) | ISO/IEC 13818-1:2000, *Information technology - Generic coding of moving pictures and associated audio information: Systems*

ISO/IEC 13818-3:1998, *Information technology - Generic coding of moving pictures and associated audio information - Part 3: Audio*

ISO/IEC 13818-7:1997, *Information technology - Generic coding of moving pictures and associated audio information - Part 7: Advanced Audio Coding (AAC)*

Contents for Subpart 5

Subpart 5: Structured Audio (SA)

5.1 Scope

5.1.1 Overview of subpart

5.1.1.1 Purpose

The Structured Audio toolset enables the transmission and decoding of synthetic sound effects and music by standardising several different components. Using Structured Audio, high-quality sound can be created at extremely low bandwidth. Typical synthetic music may be coded in this format at bitrates ranging from 0 kbps (no continuous cost) to 2 or 3 kbps for extremely subtle coding of expressive performance using multiple instruments.

MPEG-4 does not standardise a particular set of synthesis methods, but a method for describing synthesis methods. Any current or future sound-synthesis method may be described in the MPEG-4 Structured Audio format.

5.1.1.2 Introduction to major elements

There are five major elements to the Structured Audio toolset:

1. The Structured Audio Orchestra Language, or SAOL. SAOL is a digital-signal processing language which allows for the description of arbitrary synthesis and control algorithms as part of the content bitstream. The syntax and semantics of SAOL are standardised here in a normative fashion.

2. The Structured Audio Score Language, or SASL. SASL is a simple score and control language which is used in certain object types (see clause 5.6) to describe the manner in which sound-generation algorithms described in SAOL are used to produce sound.

3. The Structured Audio Sample Bank Format, or SASBF. The Sample Bank format allows for the transmission of banks of audio samples to be used in wavetable synthesis and the description of simple processing algorithms to use with them.

4. A normative scheduler description. The scheduler is the supervisory run-time element of the Structured Audio decoding process. It maps structural sound control, specified in SASL or MIDI, to real-time events dispatched using the normative sound-generation algorithms.

5. Normative reference to the MIDI standards, standardised externally by the MIDI Manufacturers Association. MIDI is an alternate means of structural control which can be used in conjunction with or instead of SASL. Although less powerful and flexible than SASL, MIDI support in this standard provides important backward-compatibility with existing content and authoring tools. MIDI support in this standard consists of a list of recognised MIDI messages and normative semantics for each.

5.2 Normative references

[DLS] MIDI Manufacturers Association , *The MIDI Downloadable Sounds Specification, v. 97.1*

[DLS2] MIDI Manufacturers Association , *The MIDI Downloadable Sounds Specification, v. 98.2*

[MIDI] MIDI Manufacturers Association , *The Complete MIDI 1.0 Detailed Specification v. 96.2*

Contents for Subpart 6

Subpart 6: Text to Speech Interface (TTSI)

6.1 Scope

This subpart of ISO/IEC 14496-3 specifies the coded representation of MPEG-4 Audio Text-to-Speech (M-TTS) and its decoder for high quality synthesized speech and for enabling various applications. The exact synthesis method is not a standardization issue partly because there are already various speech synthesis techniques.

This subpart of ISO/IEC 14496-3 is intended for application to M-TTS functionalities such as those for facial animation (FA) and moving picture (MP) interoperability with a coded bitstream. The M-TTS functionalities include a capability of utilizing prosodic information extracted from natural speech. They also include the applications to the speaking device for FA tools and a dubbing device for moving pictures by utilizing lip shape and input text information.

The text-to-speech (TTS) synthesis technology is recently becoming a rather common interface tool and begins to play an important role in various multimedia application areas. For instance, by using TTS synthesis functionality, multimedia contents with narration can be easily composed without recording natural speech sound. Moreover, TTS synthesis with facial animation (FA) / moving picture (MP) functionalities would possibly make the contents much richer. In other words, TTS technology can be used as a speech output device for FA tools and can also be used for MP dubbing with lip shape information. In MPEG-4, common interfaces only for the TTS synthesizer and for FA/MP interoperability are defined. The M-TTS functionalities can be considered as a superset of the conventional TTS framework. This TTS synthesizer can also utilize prosodic information of natural speech in addition to input text and can generate much higher quality synthetic speech. The interface bitstream format is strongly user-friendly: if some parameters of the prosodic information are not available, the missed parameters are generated by utilizing preestablished rules. The functionalities of the M-TTS thus range from conventional TTS synthesis function to natural speech coding and its application areas, i.e., from a simple TTS synthesis function to those for FA and MP.

Contents for Subpart 7

Subpart 7: Parametric Audio Coding - HILN

7.1 Scope

The Parametric Audio Coding Subpart provides the HILN tools which complement the other natural audio coding tools in the area of very low bit rates. Their focus is the representation of monophonic music signals with low and intermediate content complexity in the range of 4 to 16 kbit/s. HILN enables a high grade of interactivity by implicit support of speed and pitch change during playback and by the capability of bit rate scalability. Furthermore the possible combination with the parametric speech coding tools HVXC allows very efficient schemes for coding speech and music signals.